

# PIR schemes with small download complexity and low storage requirements

Simon R. Blackburn<sup>\*</sup>    Tuvi Etzion<sup>†</sup>    Maura B. Paterson<sup>‡</sup>

September 23, 2016

## Abstract

Shah, Rashmi and Ramchandran recently considered a model for Private Information Retrieval (PIR) where a user wishes to retrieve one of several  $R$ -bit messages from a set of  $n$  non-colluding servers. Their security model is information-theoretic. Their paper is the first to consider a model for PIR in which the database is not necessarily replicated, so allowing distributed storage techniques to be used. They concentrate on minimising the total number of bits downloaded from the servers. Shah et al. provide a construction of a scheme that requires just  $R + 1$  bits to be downloaded from servers, but requires an exponential (in  $R$ ) number of servers. We provide an improved scheme that requires a linear number of servers. Shah et al. construct a scheme with linear total storage (in  $R$ ) that needs at least  $2R$  bits to be downloaded. For any positive  $\epsilon$ , we provide a construction with the same storage property, that requires at most  $(1 + \epsilon)R$  bits to be downloaded; moreover one variant of our scheme only requires each server to store a bounded number of bits (in the sense of being bounded by a function that is independent of  $R$ ). Finally, we simplify and generalise a lower bound due to Shah et al. on the download complexity of such a PIR scheme. In a natural model, we show that an  $n$ -server PIR scheme requires at least  $nR/(n - 1)$  download bits, and provide a scheme that meets this bound.

## 1 Introduction

Shah, Rashmi and Ramchandran [11] provide bounds on the data downloaded from servers in an interesting variant of the private information retrieval model. The aim of this paper is to study this model further, improving and generalising their schemes and bounds.

---

<sup>\*</sup>Department of Mathematics, Royal Holloway University of London, Egham, Surrey TW20 0EX, United Kingdom, e-mail: [s.blackburn@rhul.ac.uk](mailto:s.blackburn@rhul.ac.uk).

<sup>†</sup>Department of Computer Science, Technion, Haifa 3200003, Israel, e-mail: [etzion@cs.technion.ac.il](mailto:etzion@cs.technion.ac.il). The research was performed while the second author visited Royal Holloway University of London under EPSRC Grant EP/N022114/1.

<sup>‡</sup>Department of Economics, Mathematics and Statistics, Birkbeck, University of London, Malet Street, London WC1E 7HX, United Kingdom, e-mail: [m.paterson@bbk.ac.uk](mailto:m.paterson@bbk.ac.uk).

## 1.1 The PIR Model

In the classical model for private information retrieval (PIR) [5], a database  $\mathbf{X}$  is replicated across  $n$  servers  $S_1, S_2, \dots, S_n$ . A user wishes to retrieve one bit of the database, so sends a query to each server and downloads their reply. The user should be able to deduce the bit from the servers' replies. Moreover, no single server should gain any information on which bit the user wishes to retrieve (without collusion). The resulting protocol is known as (an information-theoretic) *PIR scheme*; there are also computational variants of the security model. The goal of PIR is normally to minimise the total communication between the user and the servers.

The variant of this model due to Shah et al. is closer to what might be implemented in practice. They assume that each database  $\mathbf{X}$  consists of  $k$  records, each of which is  $R$  bits in length, so that the number of possible databases is  $2^{kR}$ . We denote the  $i^{\text{th}}$  record by  $R_i$ , and we write  $X_{ij}$  for the  $j^{\text{th}}$  bit of the  $i^{\text{th}}$  record. The aim of the protocol is for the user to retrieve some record  $R_j$ , rather than a single bit. Importantly, Shah et al. do not assume the whole database is replicated across the  $n$  servers  $S_1, S_2, \dots, S_n$  and so, in particular, there is the possibility of using techniques from distributed storage to reduce the total storage of the scheme. We make no restrictions on the particular encoding used to distribute the database across the servers other than to assume it is deterministic, *i.e.* that there is a unique way to encode each database. This important generalisation of the model has led to very interesting recent work which we discuss in Subsection 1.3 below.

More combinatorially, we define a private information retrieval scheme as follows.

**Definition 1.1** (PIR scheme). Suppose a database  $\mathbf{X}$  is distributed across  $n$  servers  $S_1, S_2, \dots, S_n$ . A user who wishes to learn the value of record  $R_j$  submits a *query*  $(q_1, q_2, \dots, q_n)$ . For each  $i \in \{1, 2, \dots, n\}$ , server  $S_i$  receives  $q_i$  and responds with a value  $c_i$  that depends on  $q_i$  and on the information stored by  $S_i$ . The user receives the *response*  $(c_1, c_2, \dots, c_n)$ . This system is a *private information retrieval (PIR) scheme* if the following two properties are satisfied:

**(Privacy)** For  $i = 1, 2, \dots, n$  the value  $q_i$  received by server  $S_i$  reveals no information about which record is being sought.

**(Correctness)** Given a response  $(c_1, c_2, \dots, c_n)$  to a query  $(q_1, q_2, \dots, q_n)$  for record  $R_j$ , the user is unambiguously able to recover the value of record  $R_j$ .

Note that while the query is drawn randomly according a pre-specified distribution on a set of potential queries, the response is assumed to be deterministic.

**Example 1.1.** In the case of a single server, a trivial method for achieving PIR is for the user to download the entire  $kR$ -bit database.

Chor, Goldreich, Kushilevitz and Sudan showed that in the case of single-bit records ( $R = 1$ ), if there is a single server then PIR is only possible if the total communication is at least  $k$  bits (*i.e.* the size of the entire database) [5], and so the solution above is best possible. We are interested in finding solutions such as the scheme below, in which the user downloads significantly less than  $kR$  bits from the servers:

**Example 1.2.** [5] Suppose there are two servers, each storing the entire database. Suppose  $R = 1$ .

- A user who requires record  $R_j$  chooses a  $k$ -bit string  $(\alpha_1, \alpha_2, \dots, \alpha_k)$  uniformly at random.
- Server 1 is requested to return the value  $c_1 = \bigoplus_{i=1}^k \alpha_i R_i$ , and Server 2 is requested to return  $c_2 = \left( \bigoplus_{i=1}^k \beta_i R_i \right)$ , where

$$\beta_i = \begin{cases} \alpha_i \oplus 1 & \text{when } i = j, \\ \alpha_i & \text{otherwise.} \end{cases}$$

- The user computes  $c_1 \oplus c_2$  to recover the value of  $R_j$ .

The strings  $(\alpha_1, \alpha_2, \dots, \alpha_k)$  and  $(\beta_1, \beta_2, \dots, \beta_k)$  are both uniformly distributed, and are independent of the choice of  $j$ , hence neither server receives any information as to which record is being recovered by the user.

We note that the scheme above works unchanged when the records are  $R$ -bit strings rather than single bits. The download complexity, in other words the total number of bits downloaded from servers, is  $2R$ . The following is a formalisation of the notion of download complexity used by Shah et al. [11].

**Definition 1.2.** A PIR scheme *uses binary channels* if the response  $c_j$  sent by server  $S_j$  is a binary string of length  $d_j$ , where  $d_j$  depends only on the query  $q_j$  it receives. The *download complexity* is the maximum of the sum  $\sum_{j=1}^n d_j$  over all possible queries  $(q_1, q_2, \dots, q_n)$ .

We emphasise that the length  $d_j$  in the definition above does not depend on the database  $\mathbf{X}$ , but could depend on the query  $q_j$  received by server  $S_j$ . We note that we allow for the possibility that  $d_j = 0$ , so the server does not reply to the query. Finally, we note that if we know that there are more than  $2^x$  distinct possibilities for  $c_j$  as the database varies, we may deduce that  $d_j \geq x + 1$ .

The storage requirements of a PIR scheme are of great interest:

**Definition 1.3.** Suppose server  $S_i$  stores  $s_i$  bits of information about the database  $\mathbf{X}$ .

- The *per-server storage* of the scheme is  $\max\{s_i \mid i = 1, 2, \dots, n\}$ .
- The *total storage* of the scheme is  $\sum_{i=1}^n s_i$ .

## 1.2 Results

The main results in [11] may be stated as follows:

- A proof that a PIR scheme (in the model above) must have download complexity at least  $R + 1$  when  $k \geq 2$ .
- An explicit PIR scheme that has download complexity  $R + 1$ . This scheme requires an exponential (in  $R$ ) number of servers, and so has exponential total storage.
- An explicit PIR scheme that has linear (in  $R$ ) total storage, and a download complexity of between  $2R$  and  $4R$  (so is within a constant factor of optimality).

The paper also contains the claim that a scheme of download complexity  $R + 1$  cannot have total storage that is linear in  $R$ , but no proof of this claim is given.

This paper contains analogues of, and improvements on, each of these results:

- In Section 2 we provide a more general lower bound on download complexity than the bound in [11]. In particular, the new bound implies that an  $n$ -server PIR scheme must have download complexity at least  $\frac{n}{n-1}R$  when  $k$  is sufficiently large.
- In Subsection 3.1, we provide a simple  $R + 1$ -server PIR scheme with download complexity  $R + 1$  that has total storage which is quadratic in  $R$ .
- In Subsection 3.2, we describe an  $n$ -server PIR scheme with optimal download complexity  $\frac{n}{n-1}R$ . The total storage of the scheme is linear in  $R$ . This shows that for any  $\epsilon > 0$  there exists a PIR scheme with linear total storage and download complexity at most  $(1 + \epsilon)R$ . We also describe (Subsection 3.3) a similar scheme that provides a trade-off between increasing the number of servers and reducing the per-server storage of the scheme.

### 1.3 Context

We end this introduction with a brief discussion of some of the related literature.

Private information retrieval was introduced in [5], and has been an active area ever since. See, for example, Yekhanin [17] for a fairly recent survey.

The papers by Shah et al. [11] and (independently) by Augot, Levy-Dit-Vahel, and Shikfa [1] are the first to consider PIR models where the information stored by servers could be coded using techniques from distributed storage. Whereas [11] is mainly concerned with download complexity, and also with total storage (with per-server storage, and query size also relevant parameters), the paper [1] emphasises robustness against malicious servers. The latter paper takes the total storage into account, but also emphasises other related robustness parameters: decoder locality and PIR locality.

Fazeli, Vardy, and Yaakobi [8] show how to use an object they call a PIR code (more generally a PIR array code) to provide a trade-off between the number of servers and the total storage. In particular, for all  $\epsilon > 0$ , they show that there exist good schemes (in terms of communication requirements) where the amount of information stored in a server is bounded but the total storage is at most  $(1 + \epsilon)$  times the database size. Rao and Vardy [10] study these codes further, with a lower bound on the redundancy of these PIR codes; see also Blackburn and Etzion [2].

We remark that though it is possible to reduce total storage using the techniques of PIR array codes, it seems impossible to reduce the download complexity of the resulting schemes below  $(3/2)R$  (and most codes give download complexity close to  $2R$ ) because of restrictions on the PIR rate of such codes.

Fanti and Ramchandran [6, 7] consider unsynchronized databases; the results are the same as for synchronized PIR at the expense of probabilistic success for information retrieval obtained after two rounds of communication.

Chan, Ho, and Yamamoto [3, 4] consider the tradeoff between the total storage and download complexity when the size of a record is large; the tradeoff depends on the number of records in the system.

In a sequence of papers, Sun and Jafar [12, 13, 14] consider the capacity of channels related to PIR codes in various scenarios, including the presence of colluding servers.

Finally, Tajeddine and El Rouayeb [15, 16] consider PIR schemes where the information is stored using MDS codes. They give PIR algorithms which have optimal download complexity in this model, as they attain the bounds in [3], in the situation when one or two ‘spies’ (colluding malicious servers) are present.

## 2 Lower bounds on Download Complexity

Shah, Rashmi and Ramchandran [11] show that a PIR scheme must have download complexity at least  $R+1$  when  $k \geq 2$ . Here we provide an alternative approach to proving this fact, and prove some more general results that will show our later constructions are optimal in terms of download complexity.

Throughout this section, we assume we have fixed an  $n$ -server PIR scheme, and consider its performance over all possible databases consisting of  $k$  records of length  $R$ .

**Definition 2.1.** We say that a response  $(c_1, c_2, \dots, c_n)$  is *possible* for a query  $(q_1, q_2, \dots, q_n)$  if there exists a database  $\mathbf{X}$  for which  $(c_1, c_2, \dots, c_n)$  is returned as the response to the query  $(q_1, q_2, \dots, q_n)$  when  $\mathbf{X}$  is stored by the servers.

The number of possible responses to a given query over all possible databases determines the amount of information that is downloaded by the user. This is a parameter of a PIR scheme that we would like to minimise. Similarly, we would like to minimise the size of each query, and the total amount of data stored by the servers. It is also important to consider the complexity of the computations required by both the user and the servers in carrying out a PIR scheme.

### 2.1 General bounds

We begin with a theorem that essentially shows that when a server knows that no more than  $i$  bits (where  $0 \leq i \leq R$ ) will be downloaded from the other servers, then it must reply with at least  $k(R-i)$  bits of download. Without loss of generality we will focus on server  $S_1$ , so for ease of notation we will denote the tuple  $(q_1, q_2, \dots, q_n)$  by  $(q_1, q_{\text{other}})$ , and  $(c_1, c_2, \dots, c_n)$  by  $(c_1, c_{\text{other}})$ .

**Theorem 2.1.** Suppose  $0 \leq i \leq R$ . Let  $q_1$  be fixed. Suppose we have a PIR scheme with the property that for any query of the form  $(q_1, q_{\text{other}})$ , we have

$$|\{c_{\text{other}} \mid \exists c_1 \text{ such that } (c_1, c_{\text{other}}) \text{ is possible for } (q_1, q_{\text{other}})\}| \leq 2^i.$$

Then for any query  $(q_1, q'_{\text{other}})$  we have

$$|\{c_1 \mid \exists c_{\text{other}} \text{ such that } (c_1, c_{\text{other}}) \text{ is possible for } (q_1, q'_{\text{other}})\}| \geq 2^{k(R-i)}.$$

*Proof.* Let  $q_1$  be fixed, and suppose we have a PIR scheme with the property that for any query  $(q_1, q_{\text{other}})$

$$|\{c_{\text{other}} \mid \exists c_1 \text{ such that } (c_1, c_{\text{other}}) \text{ is possible for } (q_1, q_{\text{other}})\}| \leq 2^i.$$

Assume, for a contradiction, that there exists a query  $(q_1, q_{\text{other}}^*)$  corresponding to some record  $R_j$  for which

$$|\{c_1 \mid \exists c_{\text{other}} \text{ such that } (c_1, c_{\text{other}}) \text{ is possible for } (q_1, q_{\text{other}}^*)\}| < 2^{k(R-i)}.$$

There are  $2^{kR}$  databases, and less than  $2^{k(R-i)}$  possibilities for the reply  $c_1$  of  $S_1$  to the  $(q_1, q_{\text{other}}^*)$ . So by the pigeon-hole principle, there is a value  $c_1^*$  for which there exists a set  $T$  of databases with  $|T| > 2^{kR}/2^{k(R-i)} = 2^{ki}$  having the property that for each  $\mathbf{X} \in T$ , the response of  $S_1$  to  $(q_1, q_{\text{other}}^*)$  when the servers store  $\mathbf{X}$  is  $c_1^*$ . If server  $S_1$  receives the query  $q_1$ , it will thus return  $c_1^*$  whenever a database in  $T$  is being stored.

Since the databases consist of  $k$  records, the fact that  $|T| > 2^{ki}$  implies the existence of a record  $R_\ell$  for which the number of distinct values for record  $R_\ell$  that appear among the databases in  $T$  is greater than  $2^i$ . Thus we can choose a set of databases  $W = \{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_{2^i+1}\} \subseteq T$  such that no two databases in  $W$  have the same value for record  $R_\ell$ .

The requirement for privacy against server  $S_1$  implies that if  $(q_1, q_{\text{other}}^*)$  is a query for record  $R_j$ , then there exists a query for record  $R_\ell$  of the form  $(q_1, q_{\text{other}}^\ell)$ , since otherwise  $S_1$  could distinguish between queries for  $R_j$  and queries for  $R_\ell$ . If query  $(q_1, q_{\text{other}}^\ell)$  is made when a database in  $T$  is stored, then server  $S_1$  receives  $q_1$  and responds  $c_1^*$  as before. Now consider the databases in  $W$ . As there are  $2^i + 1$  of them, yet at most  $2^i$  values for  $c_{\text{other}}$  for which there is a possible response  $(c_1^*, c_{\text{other}})$  to  $(q_1, q_{\text{other}}^\ell)$ , it follows that there must be some value  $c_{\text{other}}^\ell$  for which there are two databases  $\mathbf{X}, \mathbf{Y} \in W$  such that the response to  $(q_1, q_{\text{other}}^\ell)$  is  $(c_1^*, c_{\text{other}}^\ell)$  when either of those databases is stored. This contradicts the correctness of  $\Sigma$ , since  $\mathbf{X}$  and  $\mathbf{Y}$  do not agree in record  $R_\ell$  yet the response  $(c_1^*, c_{\text{other}}^\ell)$  to the query  $(q_1, q_{\text{other}}^\ell)$  does not allow the user to distinguish between them.  $\square$

The theorem is a generalisation of the lower bound on download complexity due to Chor et al.:

**Corollary 2.2.** [5, Theorem 5.1] *A PIR scheme that uses a single server for a database with  $k$  records of size one bit is not possible unless the number of possible responses from the server to any given query is at least  $2^k$ .*

*Proof.* Set  $i = 0$  and  $R = 1$  in Theorem 2.1.  $\square$

## 2.2 Bounds when using binary channels

Recall the definition of a PIR scheme using binary channels from the introduction. We now restrict our attention to such schemes, and provide lower bounds on the download complexity.

**Theorem 2.3.** *Let  $x$  be non-negative, and suppose we have a PIR scheme using binary channels that has total download complexity at most  $R + x$ . If the database contains  $k$  records, where  $k \geq x + 2$ , then the number of bits downloaded from any server is at most  $x$ .*

*Proof.* Without loss of generality, consider the server  $S_1$ . Suppose for a contradiction that there is a query  $q_1$  sent to  $S_1$ , where  $S_1$  replies with  $x + 1$  or more bits. Let  $i$  be the maximum number of bits downloaded by the remaining servers in reply to a query



of the form  $(q_1, q_{\text{other}})$ . Since the total download complexity is at most  $R + x$ , we find that  $0 \leq i \leq (R + x) - (x + 1) \leq R - 1$ . Since the reply  $c_{\text{other}}$  of the other servers to any query of the form  $(q_1, q_{\text{other}})$  can be expressed by a string of at most  $i$  bits, there are at most  $2^i$  possibilities for  $c_{\text{other}}$  when a query of the form  $(q_1, q_{\text{other}})$  is made. So the conditions of Theorem 2.1 are now satisfied, and we can deduce that there are at least  $2^{k(R-i)}$  possible replies  $c_1$  of  $S_1$  to the query  $q_1$ . In particular, at least  $k(R - i)$  bits are downloaded from  $S_1$ .

Let  $q'_{\text{other}}$  be chosen so that  $i$  bits are downloaded from the other servers when the query  $(q_1, q'_{\text{other}})$  is made. Then the number of bits downloaded from all servers in this situation is at least  $k(R - i) + i$ . But

$$k(R - i) + i = kR - (k - 1)i \geq kR - (k - 1)(R - 1) = R + k - 1 \geq R + (x + 2) - 1 = R + x + 1,$$

which is impossible as the scheme has total download complexity  $R + x$ . This contradiction establishes the theorem.  $\square$

**Corollary 2.4.** [11] *Let the database contain  $k$  records with  $k \geq 2$ . Any PIR scheme using binary channels requires a total download of at least  $R + 1$  bits.*

*Proof.* Suppose we have a scheme with total download of  $R$  or fewer bits. Theorem 2.3 with  $x = 0$  implies that no server downloads non-trivially, and so the user receives no information about the desired record. Hence such a scheme cannot exist.  $\square$

**Theorem 2.5.** *Suppose a PIR scheme uses binary channels and involves  $n$  servers, where  $n \geq 2$ . Suppose the database contains  $k$  records, where  $k \geq \lceil \frac{1}{n-1}R \rceil + 1$ . Then the download complexity of the scheme is at least  $\frac{n}{n-1}R$  bits.*

*Proof.* Assume for a contradiction that the scheme has download complexity  $R + x$ , where  $x$  is an integer such that  $x < \frac{1}{n-1}R$ . Since  $x \leq \lceil \frac{1}{n-1}R \rceil - 1$ , we see that  $k \geq x + 2$  and so Theorem 2.3 implies that the number of bits downloaded by any server is at most  $x$ . Since we have  $n$  servers, the total number of bits of download is always at most  $xn$ . Since our scheme has download complexity  $R + x$ , there is a query where a total of  $R + x$  bits are downloaded from servers. Hence we must have that  $nx \geq R + x$ , which implies that  $x \geq \frac{1}{n-1}R$ . This contradiction establishes the result.  $\square$

The final two results of this section concentrate on the extreme case when the download complexity is exactly  $R + 1$ .

**Corollary 2.6.** *Let the database contain  $k$  records with  $k \geq 3$ . Any PIR scheme using binary channels with a total download of exactly  $R + 1$  bits requires 1 bit to be downloaded from each of  $R$  or  $R + 1$  different servers in response to any query.*

*Proof.* The special case of Theorem 2.3 when  $x = 1$  shows that no server replies with more than 1 bit. For the download complexity to be  $R + 1$ , no more than  $R + 1$  servers can respond non-trivially. Since the user deduces the value of an  $R$ -bit record from the bits it has downloaded, at least  $R$  servers must reply.  $\square$

One might hope that the Corollary 2.6 could be strengthened to the statement that exactly  $R + 1$  servers must respond non-trivially. However, examples show that this is not always the case: see the comments after Construction 1 below.

Shah et al. state [11, Theorem 1] that, in the situation above, “for almost every PIR operation”  $R + 1$  servers must respond, and they provide a heuristic argument to support this statement. The following result makes this rigorous, with a precise definition of ‘almost every’.

**Theorem 2.7.** *Let the database contain  $k$  records with  $k \geq 3$ . Suppose we have a PIR scheme using binary channels with a total download of exactly  $R + 1$  bits. Suppose a user chooses to retrieve a record chosen with a uniform probability distribution on  $\{1, 2, \dots, k\}$ . Let  $\alpha$  be the probability that only  $R$  bits are downloaded. Then*

$$\alpha \leq \frac{R + 1}{kR + 1}.$$

*Proof.* By Corollary 2.6, each server replies to any query with at most one bit. We may assume, without loss of generality, that if a server replies with one bit then this bit must depend on the database in some way (since otherwise we may modify the scheme so that this server does not reply and the probability  $\alpha$  will increase).

Let  $(q_1, q_2, \dots, q_n)$  be a query for the record  $R_\ell$  where only  $R$  servers reply non-trivially. Since only  $R$  servers reply, there are at most  $2^R$  possible replies to the query (over all databases). But the value of  $R_\ell$  is determined by the reply, and there are  $2^R$  possible values of  $R_\ell$ . So in fact there must be exactly  $2^R$  possible replies, and there is a bijection between possible replies and possible values of  $R_\ell$ . We claim that the replies of each of these  $R$  servers can only depend on the record  $R_\ell$ , not on the rest of the database. To see this, suppose a server  $S_j$  replies non-trivially, and let  $f : \{0, 1\}^{kR} \rightarrow \{0, 1\}$  be the function mapping each possible value of the database to the reply of  $S_j$  to query  $q_j$ . Suppose  $f$  is not a function of  $R_\ell$  alone, so there are two values  $\mathbf{X}$  and  $\mathbf{X}'$  of the database that agree on  $R_\ell$  such that  $f(\mathbf{X}) \neq f(\mathbf{X}')$ . Let  $\rho$  be the common value of  $R_\ell$  in both  $\mathbf{X}$  and  $\mathbf{X}'$ . When  $R_\ell = \rho$  there are at least two possible replies to the query, depending on the value of the remainder of the database. But this contradicts the fact that we have a bijection between possible replies and possible values of  $R_\ell$ . So our claim follows.

Let  $A$  be the event that exactly  $R$  servers reply, and for  $j = 1, 2, \dots, n$  let  $B_j$  be the event that server  $S_j$  replies non-trivially. Let  $D_j$  be the indicator random variable for the event  $B_j$ . So  $D_j$  is equal to 1 when  $S_j$  responds non-trivially and 0 otherwise. Note that  $D_j$  is always equal to the number of bits downloaded from  $S_j$ , thus the expected value of the sum of these variables satisfies

$$\mathbb{E} \left( \sum_{j=1}^n D_j \right) = \alpha R + (1 - \alpha)(R + 1) = R + 1 - \alpha. \quad (1)$$

Let  $D'_j$  be the indicator random variable for the event  $A \wedge B_j$ . When  $A$  does not occur, all the variables  $D'_j$  are equal to 0. When  $A$  occurs,  $D'_j$  is the number of bits downloaded from server  $S_j$  and a total of  $R$  bits are downloaded. So

$$\mathbb{E} \left( \sum_{j=1}^n D'_j \right) = (1 - \alpha)0 + \alpha R = \alpha R. \quad (2)$$

Suppose a server  $S_j$  uses the following strategy to guess the value of  $\ell$  from the query  $q_j$  it receives. If the server replies non-trivially using a function  $f$  that depends on only



one record  $R_{\ell'}$  it guesses that  $\ell = \ell'$ . Otherwise, the server guesses a value uniformly at random. The server guesses correctly with probability  $1/k$  when it responds trivially. The argument in the paragraph above shows the server always guesses correctly if it responds non-trivially and only  $R$  servers reply. Thus the server is correct with probability at least  $(1/k) \Pr(\overline{B_j}) + \Pr(A \wedge B_j)$ . The privacy requirement of the PIR scheme implies that the server's probability of success can be at most  $1/k$ , and so we must have that  $\Pr(A \wedge B_j) \leq (1/k) \Pr(B_j)$ . Hence

$$\mathbb{E}(D'_j) \leq (1/k) \mathbb{E}(D_j).$$

By linearity of expectation, we see that

$$\mathbb{E} \left( \sum_{j=1}^n D'_j \right) = \sum_{j=1}^n \mathbb{E}(D'_j) \leq \frac{1}{k} \sum_{j=1}^n \mathbb{E}(D_j) = \frac{1}{k} \mathbb{E} \left( \sum_{j=1}^n D_j \right).$$

So, using (1) and (2), we see that

$$\alpha R \leq \frac{1}{k} (R + 1 - \alpha).$$

Rearranging this inequality in terms of  $\alpha$ , we see that the theorem follows.  $\square$

### 3 Constructions

Recall the notation from the introduction: we are assuming that our database  $\mathbf{X}$  consists of  $k$  records, each of  $R$  bits, and we write  $X_{ij}$  for the  $i$ th bit of the  $i$ th record.

#### 3.1 A scheme with download complexity $R + 1$

Shah et al. [11, Section IV] provide a PIR scheme which achieves an optimal download complexity of  $R + 1$ . However, their scheme uses an exponential (in  $R$ ) number of servers, and so has exponential total storage. The following construction, which can be thought of as a variation of the scheme of Chor et al. described in Example 1.2, achieves optimal download complexity using only  $R + 1$  servers. It has a total storage requirement which is quadratic in  $R$ .

**Construction 1.** *The following scheme is an  $R + 1$ -server PIR scheme with download complexity  $R + 1$ . All servers store the whole database.*

- A user who requires record  $R_{\ell}$  creates a  $k \times R$  array of bits by drawing its entries  $\alpha_{ij}$  uniformly and independently at random.
- Server  $R + 1$  is requested to return the bit  $c_{R+1} = \bigoplus_{i=1}^k \bigoplus_{j=1}^R \alpha_{ij} X_{ij}$ .
- For  $r = 1, 2, \dots, R$ , server  $r$  is requested to return the bit  $c_r = \bigoplus_{i=1}^k \bigoplus_{j=1}^R \beta_{ij} X_{ij}$ , where

$$\beta_{ij} = \begin{cases} \alpha_{ij} \oplus 1 & \text{if } i = \ell \text{ and } j = r, \\ \alpha_{i,j} & \text{otherwise.} \end{cases}$$

- To recover the  $r^{\text{th}}$  bit of record  $R_\ell$  the user computes  $c_r \oplus c_{R+1}$ .

We note that privacy is guaranteed since each server is asked for a uniformly random linear combination of bits from the database. The construction requires  $R + 1$  bits of download, which is optimal by Corollary 2.4. It requires  $R + 1$  copies of the database to be stored, hence has total storage of  $(R + 1)Rk$  bits. The scheme in Shah et al. requires  $(R + 1)^{k-1}$  servers, and has total storage  $(R + 1)^{k-1}R$  bits, and so the construction above is significantly better in both these metrics. However, the scheme of Shah et al. has better per server storage, as each server stores just  $R$  bits: our construction requires each server to store the whole database.

We note that there are situations where one of the servers is asked for an all-zero linear combination of bits from the database. In this case, that server need not reply. So the number of bits of downloaded in Construction 1 is sometimes  $R$  (though usually  $R + 1$  bits are downloaded). See the comment following Corollary 2.6.

In this paper, we are not aiming to optimise the number of bits uploaded to servers. Nevertheless, we provide a construction with a lower upload complexity than the schemes above. The construction can be thought of as a variant of Construction 1 where the rows of the array  $\alpha$  are all taken from a restricted set  $\{e_0, e_1, \dots, e_R\}$  of size  $R + 1$ . A similar idea is used in the constructions in [11].

For  $i = 1, 2, \dots, R$ , let  $e_i$  be the  $i^{\text{th}}$  unit vector of length  $R$ . Let  $e_0$  be the all zero vector. For binary vectors  $\mathbf{x}$  and  $\mathbf{y}$  of length  $R$ , write  $\mathbf{x} \cdot \mathbf{y}$  be their inner product; so  $\mathbf{x} \cdot \mathbf{y} = \bigoplus_{j=1}^R x_j y_j$ .

**Construction 2.** *The following scheme is an  $R + 1$ -server PIR scheme with download complexity  $R + 1$ , and reduced upload complexity. All servers store the whole database.*

- A user who requires record  $R_\ell$  chooses  $k$  elements  $a_1, a_2, \dots, a_k \in \mathbb{Z}_{R+1}$  uniformly and independently at random. For  $r = 1, \dots, R + 1$ , Server  $r$  is sent the vector  $\mathbf{b}_r = (b_{1r}, b_{2r}, \dots, b_{kr}) \in \mathbb{Z}_{R+1}^k$ , where

$$b_{ir} = \begin{cases} a_i + r \bmod R + 1 & \text{if } i = \ell, \\ a_i & \text{otherwise.} \end{cases}$$

- Server  $r$  returns the bit  $c_r = \bigoplus_{i=1}^k e_{b_{ir}} \cdot R_j$ .
- To recover the  $j^{\text{th}}$  bit of record  $R_\ell$ , the user finds the integers  $r$  and  $r'$  such that  $b_{\ell r} = 0$  and  $b_{\ell r'} = j$ . The user then computes  $c_r \oplus c_{r'}$ .

The upload complexity becomes  $(R + 1)k \lceil \log_2(R + 1) \rceil$ , significantly smaller than the upload complexity of  $(R + 1)kR$  in Construction 1.

### 3.2 Optimal download complexity for a small number of servers

For an integer  $n$  such that  $(n - 1) \mid R$ , we now describe an  $n$  server PIR scheme with download complexity  $\frac{n}{n-1}R$  bits. By Theorem 2.5, this construction provides schemes with an optimal download complexity for  $n$  servers, provided the number  $k$  of records is sufficiently large. This construction is closely related to Construction 1 above.

**Construction 3.** Let  $n$  be an integer such that  $(n - 1) \mid R$ . The following scheme is an  $n$ -server PIR scheme with download complexity  $\frac{n}{n-1}R$  bits.

- A user who requires record  $R_\ell$  creates  $R/(n - 1)$  arrays of bits, each array of size  $k \times R$ , by drawing their entries  $\alpha_{ij}^u$  uniformly and independently at random.
- Server  $n$  is asked to return the  $R/(n - 1)$ -bit string  $c_{R+1}$ , where bit  $u$  of this string is equal to  $\bigoplus_{i=1}^k \bigoplus_{j=1}^R \alpha_{ij}^u X_{ij}$ .
- For  $r = 1, 2, \dots, n - 1$ , server  $r$  is asked to return an  $R/(n - 1)$ -bit string  $c_r$ . Bit  $u$  of  $c_r$  is equal to  $\bigoplus_{i=1}^k \bigoplus_{j=1}^R \beta_{ij}^u X_{ij}$ , where

$$\beta_{ij}^u = \begin{cases} \alpha_{ij}^u \oplus 1 & \text{if } i = \ell \text{ and } j = (r - 1)(R/(n - 1)) + u, \\ \alpha_{ij}^u & \text{otherwise.} \end{cases}$$

- The user recovers the first  $R/(n - 1)$  bits of  $R_\ell$  by computing  $c_1 \oplus c_n$ , the next  $R/(n - 1)$  bits of  $R_\ell$  by computing  $c_2 \oplus c_n$  and so on.

Privacy is guaranteed since each server is asked to return  $R/(n - 1)$  independent uniformly random linear combinations of bits from the database. Each server in this construction stores the whole database, so the total storage of these schemes is  $nRk$ . Thus the total storage is linear in  $R$ .

Shah et al. [11, Section V] provide PIR schemes with linear (in  $R$ ) total storage and with download complexity between  $2R$  and  $4R$ . Their scheme requires a number of servers which is independent of  $R$  (but is linear in  $k$ ). The construction above (taking  $n$  to be fixed but sufficiently large) shows that for any fixed positive  $\epsilon$  a PIR scheme with linear total storage exists with download complexity of  $(1 + \epsilon)R$ : this is within an arbitrarily close factor of optimality. Moreover, the number of servers in our construction is independent of both  $k$  and  $R$ . However, note that in our scheme each server stores the whole database, whereas the per server storage of the scheme of Shah et al. is a fixed multiple of  $R$ .

### 3.3 Schemes with small per-server storage

We make the observation that the last construction may be used to give families of schemes with lower per-server storage; see [11, Section V] for similar techniques.

**Construction 4.** Let  $s$  be a fixed integer such that  $s \mid R$ . Let  $r$  be a fixed integer such that  $(r - 1) \mid s$ . Let  $n = r(R/s)$ . The following scheme is an  $n$ -server PIR scheme with download complexity  $\frac{r}{r-1}R$  bits.

- Divide each record into  $R/s$  chunks of  $s$  bits each. Divide the database into  $R/s$  parts of  $ks$  bits, the  $i$ th part containing the  $i$ th chunk of each record.
- Operate  $R/s$  copies of the PIR scheme of Construction 3 independently. Each copy uses  $r$  servers; no server is used in two copies of the scheme. The  $i$ th copy of the scheme operates on the  $i$ th part of the database only (and so each server needs to store just one part of the database).

The download complexity of this PIR scheme is  $\frac{R}{s} \frac{r}{r-1} s = \frac{r}{r-1} R$  bits. Each server stores just  $ks$  bits. The total storage requirements of the scheme is  $nks = rkR$ . By fixing  $r$  and  $s$  to be sufficiently large integers, we can see that for all positive  $\epsilon$  we have a family of schemes with download complexity at most  $(1 + \epsilon)R$ , with total storage linear in the database size, with a linear (in  $R$ ) number of servers, and where the per server storage is independent of  $R$ . So this family of schemes has a better download complexity and per-server storage than Shah et al. [11, Section V], and is comparable in terms of both the number of servers and total storage.

We remark that the above construction still works if the sets of  $r$  servers are not disjoint: the storage requirements of those servers in more than one  $r$ -set is increased, but the download complexity and total storage are unaffected and the number of servers required is reduced. So various trade-offs are possible using this technique.

## References

- [1] D. AUGOT, F. LEVY-DIT-VAHEL, AND A. SHIKFA, *A storage-efficient and robust private information retrieval scheme allowing few servers*, in *Cryptology and Network Security*, 222–239, Springer 2014.
- [2] S.R. BLACKBURN AND T. ETZION, *PIR array codes with optimal PIR rate*, [arxiv.org/abs/1607.00235](https://arxiv.org/abs/1607.00235), August 2016.
- [3] T. H. CHAN, S. HO, AND H. YAMAMOTO, *Private information retrieval for coded storage*, [arxiv.org/abs/1410.5489](https://arxiv.org/abs/1410.5489), October 2014.
- [4] T. H. CHAN, S. HO, AND H. YAMAMOTO, *Private information retrieval for coded storage*, *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT)*, pp.2842–2846, Hong Kong, June 2015.
- [5] B. CHOR, O. GOLDBREICH, E. KUSHILEVITZ, AND M. SUDAN, *Private information retrieval*, *Journal ACM*, pp.065–981, 1998.
- [6] G. FANTI AND K. RAMCHANDRAN,, *Multi-server private information retrieval over unsynchronized databases*, *Fifty-second Annual Allerton Conference*, pp.437–444, Illinois, October 2014.
- [7] G. FANTI AND K. RAMCHANDRAN, *Efficient private information retrieval over unsynchronized databases*, *IEEE J. on Selected Topics in Signal Processing*, 9 (2015), 1229–1239.
- [8] A. FAZELI, A. VARDY, AND E. YAAKOBI, *Coded for distributed PIR with low storage overhead*, *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT)*, pp.2852–2856, Hong Kong, June 2015.
- [9] A. FAZELI, A. VARDY, AND E. YAAKOBI, *Private information retrieval without storage overhead: coding instead of replication*, [arxiv.org/abs/1505.06241](https://arxiv.org/abs/1505.06241), May 2015.
- [10] S. RAO AND A. VARDY, *Lower bound on the redundancy of PIR codes*, [arxiv.org/abs/1605.01869](https://arxiv.org/abs/1605.01869), May 2016.

- [11] N.B. Shah, K.V. Rashmi and K. Ramchandran, ‘One extra bit of download ensures perfectly private information retrieval’, in *Proc. of IEEE Int. Symp. on Inform. Theory (ISIT)*, pp. 856–860, Honolulu, June 2014.
- [12] H. SUN AND A. JAFAR, *The capacity of private information retrieval*, *arxiv.org/abs/1602.09134*, February 2016.
- [13] H. SUN AND A. JAFAR, *The capacity of robust private information retrieval with colluding databases*, *arxiv.org/abs/1605.00635*, May 2016.
- [14] H. SUN AND A. JAFAR, *The capacity of symmetric private information retrieval*, *arxiv.org/abs/1606.08828*, June 2016.
- [15] R. TAJEDDINE AND S. EL ROUAYHEB, *Private information retrieval from MDS coded data in distributed storage systems*, *arxiv.org/abs/1602.01458*, February 2016.
- [16] R. Tajeddine and Salim El Rouayheb, “Private information retrieval from MDS coded data in distributed storage systems,” *Proc. IEEE International Symposium on Information Theory*, pp. 1411–1415, Barcelona, Spain, 2016.
- [17] S. YEKHANIN, *Private information retrieval*, *Comm. ACM*, 53 (2010), 68–73.